

# Model-Based Approach to Accurate and Consistent 3-D Modeling of Drainage and Surrounding Terrain

P. Fua\*

Computer Graphics Lab  
EPFL  
CH-1015 Lausanne

AI Center  
and SRI International  
CA 94025 Menlo Park

## Abstract

*We propose an automated approach to modeling drainage channels—and, more generally, linear features that lie on the terrain—from multiple images, which results not only in high-resolution, accurate and consistent models of the features, but also of the surrounding terrain.*

*In our specific case, we have chosen to exploit the fact that rivers flow downhill and lie at the bottom of local depressions in the terrain, valley floors tend to be “U” shaped, and the drainage pattern appears as a network of linear features that can be visually detected in single gray-level images.*

*Different approaches have explored individual facets of this problem. Ours unifies these elements in a common framework. We accurately model terrain and features as 3-dimensional objects from several information sources that may be in error and inconsistent with one another. This approach allows us to generate models that are faithful to sensor data, internally consistent and consistent with physical constraints.*

## 1 Introduction

We propose an automated approach to modeling drainage channels—and, more generally, linear features that lie on the terrain—from multiple images, which results not only in high-resolution, accurate and consistent models of the features, but also of the surrounding terrain.

This is an important problem from both a practical point of view—drainage modeling is an essential component of map making—and a theoretical point of view: We must address two key generic problems. The first is the obvious requirement to replace reliance on generally unavailable prior knowledge of explicit shape with more general ways of re-

cognizing and describing natural objects. The second is the necessity to merge several sources of information that may not be consistent with one another.

Different approaches have explored individual facets of this problem. There is extensive literature on the extraction of valleys from terrain models, for example see [2] among many others. The terrain model, however, is almost always assumed to be error-free, which, in practice, only rarely is the case. Furthermore, Koenderink and Van Doorn have shown [9] that the local differential criteria many of these systems use to detect valleys have inherent problems. Much work has also been devoted to the extraction of linear patterns from single images using techniques such as dynamic programming [4, 10] or graph-based techniques [3]. These techniques typically do not use the terrain information or guarantee that the recovered drainage pattern satisfies the physical constraints discussed above. Furthermore they do not take advantage of the fact that multiple images of the same site may be available.

Our approach unifies these elements in a common framework. Because the features and the physical constraints we deal with are fundamentally 3-D and because we want to be able to deal with an arbitrary number of images, there are very significant advantages in using an object-centered 3-D representation of the terrain surface and features that allows us to effectively enforce consistency constraints.

We have chosen to concentrate on the extraction and the refinement of drainage patterns because they are potentially complex but obey well-understood physical constraints and therefore constitute a very good test case for our research. However, as discussed in an extended version of this paper [5], the same techniques are robust enough to work on other linear features that are constrained by predictable forces such as roads and ridgelines.

We view the contribution of this paper as proposing a general approach to accurately modeling terrain and features from several information sources that may be in error and inconsistent with one another. This approach allows us to generate models that are faithful to sensor data, internally

---

\*This work was supported in part by contracts from the Defense Advanced Research Projects Agency.

consistent and consistent with physical constraints.

We first introduce our overall framework. We then review our approach to modeling the terrain and estimating its curvature and present the techniques we use to quickly sketch the drainage pattern and to automatically enforce the consistency constraints.

## 2 Approach

We model the terrain as a triangulated mesh that can be refined by minimizing an objective function, and we model the rivers' locations as polygonal paths that lie on the terrain surface, and are located where the largest principal curvature of the terrain surface is locally maximal in the direction normal to the path. Furthermore, their tangent vectors are the directions of maximal elevation decrease and their altitude decreases monotonically.

We start by recovering the approximate shape of a terrain mesh by minimizing a multi-image stereo score [7] and computing a curvature map. From this map, we extract paths of maximal curvature using dynamic programming. This simple approach would be sufficient if the recovered terrain surface was perfect and if the terrain was steep enough for all streams to flow at the bottom of the sort of "V" shaped valleys that erosion produces. In practice, this is not always the case. There may not be enough relief to tell the real but shallow valleys from spurious valleys that may be present in the recovered terrain surface. Furthermore, even if there are deep and easy-to-detect "V" shaped valleys, vegetation tends to be taller on river banks, thus making the elevations computed by our surface-reconstruction algorithm unreliable. As a result, the recovered path may not follow the true valley bottom and may not exhibit monotonically decreasing elevations.

To solve these problems, we have developed a more sophisticated approach:

- We use both the terrain model and the actual gray-level images to extract a rough estimate of the features' locations, thus preventing the estimate from being too far off if there are errors in the terrain model.
- We simultaneously refine the models of the terrain and features under consistency constraints that ensure that they fit the image data as well as possible while conforming to the physical constraints known to apply.

We will show that this technique allows the quick generation of accurate and consistent 3-D models of the drainage channels and the surrounding terrain with minimal manual intervention.

## 3 Terrain Modeling

In this work, we represent the terrain as regular 3-D triangulated mesh that can be projected into an arbitrary number of images, thus allowing us to integrate information from these images, to model surfaces of arbitrary orientations and to impose geometric constraints [7].

### 3.1 Recovering the Shape of the Terrain

The shape of a terrain mesh  $S$  is defined by the position of its vertices. It can be refined by minimizing a regularized objective function that accounts for the stereo information present in multiple images of a cartographic site to produce models such as the one shown in Figure 1(c). In other words, this technique uses 3-D triangulations not only as a representational tool but also as a computational one.

The objective function  $\mathcal{E}(S)$  is taken to be

$$\mathcal{E}(S) = \mathcal{E}_D(S) + \mathcal{E}_{St}(S) , \quad (1)$$

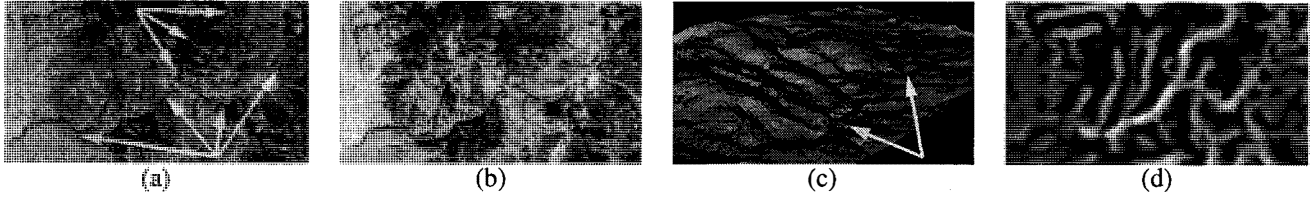
where  $\mathcal{E}_D(S)$  is a regularization term that is quadratic in terms of the vertices' coordinates and  $\mathcal{E}_{St}(S)$  is a multiple-image correlation term [7]. This method allows us to deal with arbitrarily slanted regions and to discount occluded areas of the surface.

In our application, we fix the  $x$  and  $y$  coordinates of the vertices of  $S$ , and the free variables are the  $z$  coordinates. The process is started with an initial estimate of the elevations typically derived from a coarse DEM. In the course of the optimization, we progressively refine the mesh by iteratively subdividing the facets into four smaller ones whose sides are still of roughly equal length, thus preserving its regularity.

### 3.2 Differential Properties of the Terrain

In Section 4, we will show that we can combine the differential properties of the terrain surface—specifically, its largest principal curvature—with the information present in the gray-level images to automate the delineation of the drainage pattern. It is therefore important to be able to represent both kinds of information in a common frame of reference. In our application, we deal with near-vertical aerial imagery and we either use an ortho photo or the vertical-most available image.

Following Sander and Zucker [12], we estimate the maximal curvature at each vertex of the surface by fitting a quadric to the vertices in the neighborhood of that vertex. For each point on the surface, we then use a weighted average of the curvatures of the three vertices of the facet to which it belongs.



**Figure 1.** Terrain modeling at the National Training Center (NTC), Ft. Irwin. (a,b) A stereo pair of a hilly site. The linear structure that runs horizontally in the middle of the images is a streambed and is indicated by the cluster of arrows in (a). The other cluster of arrows denotes a second streambed that runs vertically. (c) A shaded view of the terrain mesh recovered by our system after optimization. The two arrows point at the valley of the horizontal streambed shown in (a). The second streambed is hidden behind one of the hills. (d) The curvature image registered to the image shown in (a). Regions of high positive curvature—that is, candidate valley regions—are shown in white.

Using this method and given a surface triangulation, we can compute, for each original gray-level image, a “curvature image” that is registered with it such as the ones shown in Figure 1(d).

## 4 Automating Drainage Delineation

We distinguish between steep terrain where the geometry of the terrain surface is usually sufficient to detect the drainage channels and less steep terrain where geometry becomes less relevant and the information present in the original images must be used more directly.

### 4.1 Steep Terrain

In high-relief areas, rivers create valleys by eroding the surrounding terrain and over time carve channels that typically are not completely filled. As a result they tend to appear as local depressions and their center lines closely match maxima of curvature in the terrain surface.

It is therefore natural to look for paths of maximum curvature in the “curvature images” introduced in Section 3.2 and computed using the terrain mesh. As shown in Figure 2, this can be achieved by simply specifying endpoints and using a fast dynamic programming algorithm [11, 1] to find—in real time—a path  $\mathcal{C}$  that minimizes

$$\mathcal{E}_{Curv}(\mathcal{C}) = \int (C_{max}(\mathbf{f}(s)) - C_{max}^1)^2 ds, \quad (2)$$

where  $\mathbf{f}(s)$  is a vector function mapping the arc length  $s$  to points  $(u, v)$  along the curve,  $C_{max}(u, v)$  is the terrain’s surface maximal curvature at image location  $(u, v)$ , and  $C_{max}^1$  is the largest value of  $C_{max}(u, v)$  in the curvature image. It can be shown [5], that if  $\mathcal{C}$  minimizes  $\mathcal{E}_{Curv}(\mathcal{C})$ , it is close to being the locus of points that are maxima of curvature in the direction normal to the curve. We therefore refer to these paths as “maximal curvature” paths.

### 4.2 Flat Terrain

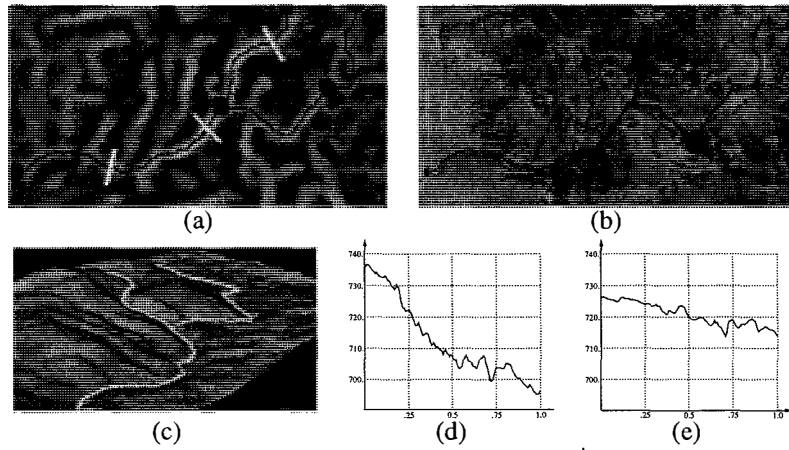
As the terrain’s relief becomes less pronounced, the channels become increasingly difficult to detect from the geometry of the surface mesh alone. In the limit, a river meandering through an almost flat flood plain could not be sketched using the technique described above.

Figure 3 illustrates this problem in an area where the terrain’s shape is close to that of a slanted plane. In such cases the clues to the river’s presence are to be found in the original gray-level images where they appear as elongated linear structures that can be detected using a low-resolution linear delineation system such as the one developed by Fischler and Wolf [4]. It yields results such as those shown in Figure 3(e). These linear features can then be chamfered and used to mask the curvature image. This operation produces the image of Figure 3(f) in which the curvature of all points but those that are close to one of the linear structures is set to  $C_{max}^0$ , the smallest value in the original curvature image, and will therefore tend to be avoided by the dynamic programming algorithm. Using the same endpoints as previously, we obtain the paths shown in Figure 3(g) that are much closer to those that a human analyst would delineate using a stereoscope.

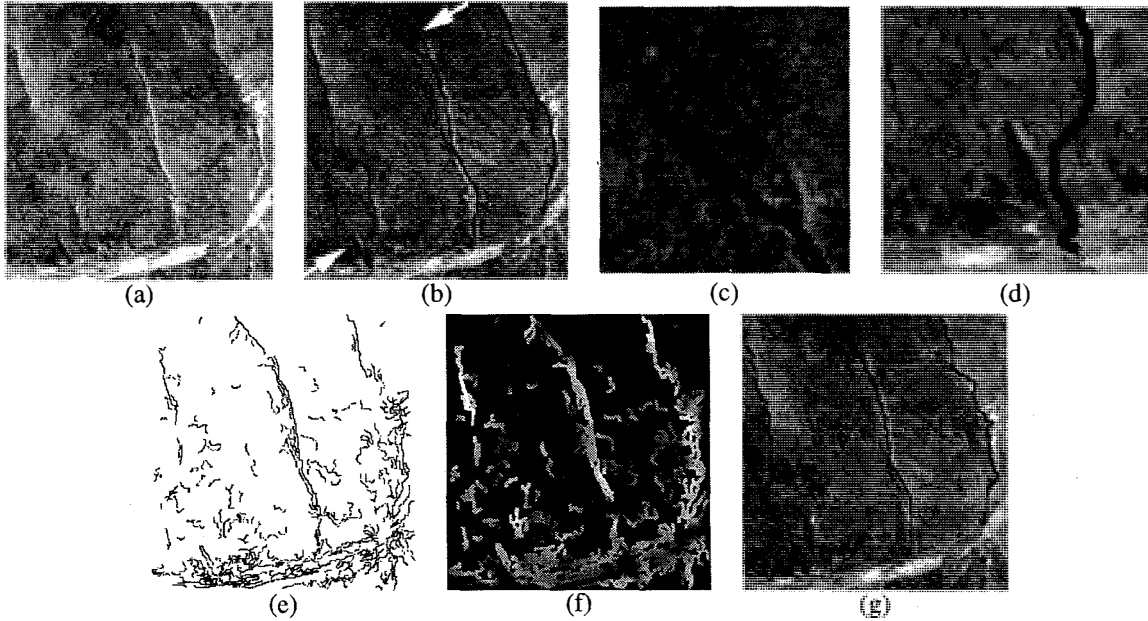
## 5 Enforcing the Physical Constraints

We now turn to the physical constraints that the drainage pattern and surrounding terrain must fulfil. As illustrated by Figure 2, there is no guarantee that the features sketched using the techniques of Section 4 will be consistent with the laws of physics because the terrain model may be in error.

Our goal is therefore to enforce these constraints while deviating as little as possible from what the image data predicts; otherwise we might be “hallucinating” river valleys where there are none. Constrained optimization is an effective way to achieve this goal because it allows the use of arbitrarily large numbers of constraints while retaining good



**Figure 2.** Sketching the rivers at the NTC site. (a) The maximal curvature paths overlaid on the curvature image of Figure 1(f). For one of the paths, we specified two endpoints and one intermediate point denoted by the black circles; for the other we specified only the two endpoints denoted by the black rectangles. (b) The paths overlaid on the original image. (c) The paths overlaid on a shaded view of the terrain mesh of Figure 1(e). (d,e) Elevations along the paths. Note that because of imprecisions in the reconstruction, they are not monotonic.



**Figure 3.** Drainage delineation in flatter terrain. (a) One of three images of an area where the terrain is close to being a slanted plane with the highest elevations at the top of the image. Note the three gullies running from top to bottom of the image. (b) The maximal curvature paths computed by specifying two endpoints for each gully. (c) Detail of the upper part of the middle path—denoted by the topmost white arrow in (b)—that meanders away from the clearly visible linear structure that marks the actual location of the gully. (d) Similar problem in the lower part of the leftmost path, denoted by the other white arrow in (b). (e) The linear features detected by the low-resolution linear delineation system in the image of (a). (f) The potential image computed by using those linear features to mask the curvature image. (g) The linear structures delineated by using the potential image (f) and supplying two endpoints for each of the three linear structures.

convergence properties. In fact, the more constraints there are, the smaller the search space and the better the convergence becomes [6].

## 5.1 Constrained Optimization

Formally, a constrained optimization problem can be described as follows. Given a function  $f$  of  $n$  variables

$S = \{s_1, s_2, \dots, s_n\}$ , we want to minimize it under a set of  $m$  constraints  $C(S) = \{c_1, c_2, \dots, c_m\} = 0$ . That is,

$$\text{minimize } f(S) \text{ subject to } C(S) = 0 \quad (3)$$

It can be generalized to handle inequality constraints by replacing the constraints of the form  $c_i(S) = 0$  by constraints of the form  $c_i(S) \leq 0$ .

In our application, we model the terrain as a triangulated mesh  $\mathcal{S}$  and linear features as a set of  $l$  polygonal curves  $\mathcal{C}_{j, 1 \leq j \leq l}$ . We associate to each an energy term  $\mathcal{E}(\mathcal{S})$  and  $\mathcal{E}(\mathcal{C}_j) - \mathcal{E}(\mathcal{S})$  is discussed in Section 3 and  $\mathcal{E}(\mathcal{C}_j)$  is introduced below.  $S$  is therefore the vector of all the  $x, y$  and  $z$  coordinates of the vertices of  $\mathcal{S}$  and of the  $\mathcal{C}_j$ s.  $f(S)$  is taken to be

$$f(S) = \{\mathcal{E}(\mathcal{S}), \mathcal{E}(\mathcal{C}_1), \dots, \mathcal{E}(\mathcal{C}_l)\} ,$$

and our algorithm minimizes each component of  $f$  while attempting to satisfy the constraints.

We must now express the fact that rivers flow downhill and lie at the bottom of local depressions in the terrain and that valley floors tend to be “U” shaped and locally horizontal in the direction transverse to the river’s direction in terms of a set of constraints of the form  $c_i(S) = 0$  or  $c_i(S) \leq 0$ :

**Rivers lie at the bottom of valleys:** We treat a river as smooth 3-D curve  $\mathcal{C}$ . We refine its position by minimizing an energy  $\mathcal{E}(\mathcal{C})$  that is the weighted sum of a regularization term  $\mathcal{E}_D(\mathcal{C})$ —the integral of the square curvatures along the curve—and a potential term  $\mathcal{E}_P(\mathcal{C})$ —minus the integral of the elevations along the curve.

In practice, following standard snake practices [8], we model  $\mathcal{C}$  as a list of regularly spaced 3-D vertices  $\mathcal{S}_3$  of the form  $\mathcal{S}_3 = \{(x_i, y_i, z_i), i = 1, \dots, n\}$ , and we discretize  $\mathcal{E}_D$  and  $\mathcal{E}_P$ .

As discussed below, during the optimization the curve  $\mathcal{C}$  is constrained to remain on the terrain and while the vertices are moved to minimize  $\mathcal{E}_P(\mathcal{C})$  and therefore the elevations of the individual vertices. As a result, the curve has to lie at the bottom of a valley.

**Rivers flow downhill:** The  $z$  coordinates of the curve’s list of  $n$  3-D vertices  $\mathcal{S}_3$  decrease monotonically, which is expressed as a set of  $n - 1$  inequality constraints  $z_{i+1} \leq z_i$ , that we refer to as “downhill” constraints.

#### Rivers lie on the terrain:

For each edge  $((x_1, y_1, z_1), (x_2, y_2, z_2))$  of the terrain mesh and each segment  $((x_3, y_3, z_3), (x_4, y_4, z_4))$  of the polygonal curve representing the river that intersect when projected in the  $(x, y)$  plane, the four endpoints must be coplanar so

that the segments also intersect in 3-D space. This is written as

$$\begin{vmatrix} x_1 & x_2 & x_3 & x_4 \\ y_1 & y_2 & y_3 & y_4 \\ z_1 & z_2 & z_3 & z_4 \\ 1 & 1 & 1 & 1 \end{vmatrix} = 0 ,$$

which yields a set of constraints that we refer to as “on-terrain” constraints.

**A valley is horizontal in the direction perpendicular to the river’s direction:** Each edge  $((x_1, y_1, z_1), (x_2, y_2, z_2))$  of the terrain mesh that intersects, in the  $(x, y)$  plane, a segment  $((x_3, y_3, z_3), (x_4, y_4, z_4))$  must be in the plane defined by the segment and the horizontal vector normal to it. This can be written as

$$\begin{vmatrix} x_2 - x_1 & x_4 - x_3 & y_3 - y_4 \\ y_2 - y_1 & y_4 - y_3 & x_4 - x_3 \\ z_2 - z_1 & z_4 - z_3 & 0 \end{vmatrix} = 0 ,$$

This yields another set of constraints that we refer to as “valley-bottom” constraints.

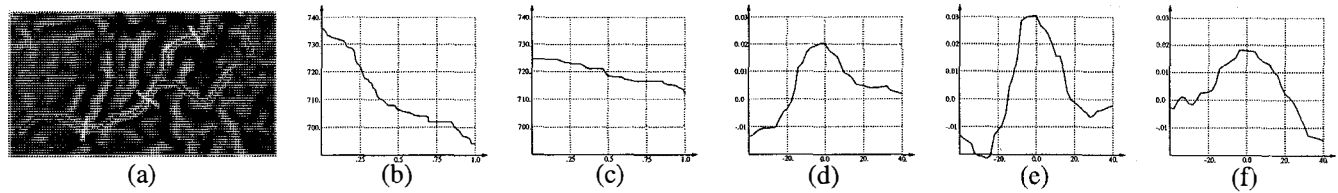
The downhill constraints are inequality constraints that are turned on and off during the optimization as required, following an active set strategy [6]. Before the start of the constrained optimization, we compute the intersections in the  $x, y$  plane between the edges of the mesh and the polygonal curves to instantiate the required number of on-terrain and valley-bottom constraints. Optionally, we could reiterate this procedure during the optimization. This would be necessary if the polygonal curves deformed a lot. However, because the delineation method of Section 4 is robust, the initial location of the curves is accurate enough so that, in practice, they do not deform very much.

Figure 4 demonstrates the improvement in consistency brought about by constrained optimization: The channels now have monotonically decreasing elevations and the rivers lie at their bottoms, which also is close to being a maximum of curvature in the direction normal to the feature.

## 6 Conclusion

We have presented an approach to terrain modeling and 3-D linear delineation that allows us to generate site models including terrain drainage channels and roads that are accurate and consistent with minimal human intervention.

We have shown that, by refining an object-centered representation of the terrain and features under a set of well-designed constraints, we can generate, with a high level of automation, models that are faithful to sensor data, internally consistent and consistent with physical constraints. We have also shown that we can achieve this result in a highly automated fashion: the operator is only required to



**Figure 4.** NTC model after constrained optimization. (a) The optimized streambeds of Figure 2 overlaid on a recomputed curvature image. (b,c) Their elevations are now monotonically decreasing, unlike those of Figures 2. (d,e,f) Curvature of the surface along the three perpendicular cross sections shown as white segments in (c).

specify a few endpoints, and the system handles everything else.

We have concentrated on the modeling of drainage patterns but the framework described here extends naturally to modeling all objects obeying known physical constraints. For example, man-made objects such as roads, railroad tracks, or buildings are built according to well-understood engineering practices. Similarly, silhouette edges can be extracted from ground-level views of mountain ridges and used to constrain the terrain modeling from aerial views.

We believe that the capabilities described here will prove indispensable to automating the generation of complex object databases from imagery, such as the ones required for realistic simulations or intelligence analysis. In such databases, the models must not only be as accurate—that is, true to the data—as possible but also consistent with each other. Otherwise, the simulation will exhibit “glitches” and the image analyst will have difficulty interpreting the models.

## Acknowledgements

We wish to thank Martin Fischler and Yvan Leclerc for the invaluable advice they provided during the development of the techniques described here. We also would like to thank Lee Iverson for providing us with a real-time implementation of the dynamic programming algorithm; it has proved to be a powerful tool for experimenting with different approaches to generating the cost arrays of Section 4.

## References

- [1] L. Cohen and R. Kimmel. Global Minimum for Active Contour Models: A Minimal Path Approach. In *Conference on Computer Vision and Pattern Recognition*, pages 666–673, San Francisco, CA, June 1996.
- [2] J. Fairfield and P. Leymarie. Drainage Networks from Grid Digital Evaluation Models. *Water Resources Research*, 27(5):709–717, May 1991.
- [3] M.A. Fischler, J.M. Tenenbaum, and H.C. Wolf. Detection of Roads and Linear Structures in Low-resolution Aerial Imagery Using a Multisource Knowledge Integration Technique. *Computer Vision, Graphics, and Image Processing*, 15(3):201–223, March 1981.
- [4] M.A. Fischler and H.C. Wolf. Linear Delineation. In *Conference on Computer Vision and Pattern Recognition*, pages 351–356, June 1983.
- [5] P. Fua. Fast, Accurate and Consistent Modeling of Drainage and Surrounding Terrain. *International Journal of Computer Vision*, 1997. Accepted for publication, available as Tech Note 555, Artificial Intelligence Center, SRI International.
- [6] P. Fua and C. Brechbühler. Imposing Hard Constraints on Soft Snakes. In *European Conference on Computer Vision*, pages 495–506, Cambridge, England, April 1996. Available as Tech Note 553, Artificial Intelligence Center, SRI International.
- [7] P. Fua and Y. G. Leclerc. Object-Centered Surface Reconstruction: Combining Multi-Image Stereo and Shading. *International Journal of Computer Vision*, 16:35–56, September 1995.
- [8] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active Contour Models. *International Journal of Computer Vision*, 1(4):321–331, 1988.
- [9] J.J. Koenderink and J. van Doorn. Local Features of Smooth Shapes: Ridges and Courses. In *SPIE*, volume 2031, 1993.
- [10] N. Merlet and J. Zerubia. New Prospects in Line Detection by Dynamic Programming. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(4), April 1995.
- [11] E.N. Mortensen and W.A. Barrett. Intelligent Scissors for Image Composition. In *Computer Graphics, SIGGRAPH Proceedings*, pages 191–198, Los Angeles, CA, August 1995.
- [12] P.T. Sander and S.W. Zucker. Inferring Surface Trace and Differential Structure from 3-D Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(9):833–854, September 1990.